## Unit 3: Problems

**Problem 1**

We have a vector V [1..N] formed by integers, so that all of them are different, and that they are ordered in an increasing way. It is said that a vector of these characteristics is coincident if it has at least one position such that it is equal to the value that contains the vector in that position. For example, in the vector

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| -14 | -6 | 3 | 6 | 16 | 18 | 27 | 43 |

it can be seen that V [3] = 3; therefore, this vector is coincident.

Design a Divide and Conquer algorithm that determines in an order of efficiency not greater than O(logn) if a vector is coincident, receiving as data the vector and its size.

**Problem 2**

We have a vector V [1..N] from which we want to obtain information as if it were ordered, but that can not be ordered for work reasons (nor can copies be made in memory of the vector). To do so, we want to create an index vector Ind[1..N] in such a way that the value Ind[i] is precisely the position of the original vector with the data that should be in position i if it were indeed ordered. For example, if the vector V[1..N] was

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 50 | 98 | 10 | 63 | 31 | 25 | 63 | 74 |

then the index vector Ind [1..N] would be

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| 3 | 6 | 5 | 1 | 4 | 7 | 8 | 2 |

since Ind[1] = 3 is the position of V where 10 is, which is the element that would be first if V were ordered, Ind[2] = 6 is the position of 25, which would be the second element in the ordered vector , etc.

Modify the merging algorithm by Mergesort to obtain the vector of indices of a vector V[1..N] given as a parameter.

**Problem 3**

Consider a function f(x) which is known to have a unique local minimum at point x0 in the real interval [p1, p2], which is strictly decreasing between [p1, x0] and which is strictly increasing between [x0, p2]. Note that x0 may coincide with p1 or with p2.

Search, as efficiently as possible, all the points x (if they exist) of the interval [p1, p2] such that the function f takes a certain value k, that is, you look for the set of values $\{x \in [p1, p2]$ such that $f(x) = k\}$. To simplify the process, instead of the exact value of each x, a range of values $[\alpha, \beta]$ can be indicated, with $\beta - \alpha < \varepsilon$, where x is found. The algorithm data will be the interval [p1, p2], the value k that is being searched for, and the value $\varepsilon$ for the approximation.

**Problem 4**

You want to program a robot to put cork stoppers to the bottles of a recycling factory. There are available N bottles and the N corks that cover them (N is constant in the problem), but there are a number of problems:

- The bottles are all different from each other, just like corks: each bottle can only be closed with a concrete cork, and each cork only serves for a specific bottle.
- The robot is prepared to close bottles, so all he knows how to do is compare corks with bottles. The robot can detect if a cork is too small, too large, or just the right size to close a bottle.
- The robot can not compare corks with each other to "sort" them by thickness, nor can it do so with the bottles.
- The robot has space available and mechanical arms to place bottles and corks at will, for example in different positions of a table, if necessary.

Design the algorithm that the robot needs to plug N bottles optimally.

**Problem 5**

In Aceland the national sport is tennis. There is a ranking, in which each player is assigned a number of points according to their quality, that is, the best player in that country is the one with the highest number of points. Every year a couple must be selected from among all the Aceland players to play a doubles tournament at an international level. The selection procedure is a little peculiar. The score of each of the players is placed on a list, in a completely random way, without any sort of order. Once the list is made, each player can only form a pair with an adjacent player on the list, that is, who is in front of or behind him on that list. Obviously, the first player on the list can only pair with the second and the last with the penultimate, but the rest have two possible options to form the pair of doubles, corresponding to the previous and subsequent players on the list. With this restriction, the best pair of doubles possible is chosen, which is the one in which the sum of the points of its two components is greater. Design an algorithm whose main function follows the divide and conquer scheme, which decides which pair of doubles should compete in Aceland. Reason the complexity of the algorithm.

**Problem 6**

After passing through the Tile Room and getting the Cradle of Life, now Indiana Croft faces a new challenge before being able to leave the Temple Cursed. It is located on a bridge under which there is deep darkness.

Fortunately, this place also appears in the diary. The bridge crosses the so-called Valley of Shadows, which begins with a descent slope (the slope is not necessarily constant) so after reaching the lowest point start to climb to the other end of the bridge (again, not necessarily with constant slope). Just at the bottom of the valley there is a river, but the diary does not specify its location with respect to the bridge (for example, it is not known if the river is 53 meters from the beginning of the bridge) or the distance in height (ie , it is not known if the river is 228 meters below, for example). On the slopes there are sharp rocks.

If Indiana Croft had time, he could easily find the point where he can get off the bridge to get exactly to the river, as it has a laser pointer to measure heights that tells him how many meters there are from the bridge to the ground at a certain point. The problem is that the priests of the temple have found out that they have stolen the Cradle of Life, they are chasing Indiana Croft and they will reach him right away. The adventurer must quickly find the position of the river to get off and flee safely.

Design the algorithm that Indiana Croft should use to find the minimum point of the valley under the indicated conditions. The algorithm must be efficient: at least in the best case it must have a logarithmic order. You can consider the time it takes for Indiana Croft to travel along the bridge is null and that the estimate of the point of the river where to drop off can have an approximation error of ε meters (ε is a given constant).

**Problem 7**

In Abeceland, a city famous for its N beautiful squares and that maybe you know, they have a curious system of roads: from each square a street goes to all the other squares that begin with a letter that is in their name (for example, from Ace, there are streets that lead to the squares that begin with C, as Cut or Coast Squares, or by E, as East Square, but do not have streets to places like Doom, Fall or Tiara). The streets are one-way (Ace Square can go to Cut Square, but not the other way around since it does not meet the rule of the letters, obviously, other places like Ace and Cat are connected to each other in both directions). All these connections between the N squares are collected in a street map, represented by an adjacency matrix of size NxN; thus, the value of Streets[p, q] indicates whether one can go from plaza p to plaza q.

April 26, festival of San Isidoro de Sevilla (patron of letters and, coincidentally, computer scientists) is approaching, and the City of Abeceland have decided to celebrate it reversing the address of all streets that connect their places. On that day you can not move from Ace to Cut, but you will be allowed to go from Cut to Ace to Aro; Obviously, Ace and Cat will remain connected to each other.

To formally design a standard Divide and Conquer algorithm that, having as data the street map of the city (represented by the adjacency matrix), obtain the new street map valid for the day of San Isidoro of Seville, indicating the data structures that are used.

**Deliverables:** An exercise to choose among problems 3 and 4 and an exercise to choose among problems 6 and 7.